What Is Claimed Is:

1. A method for precise feedback data generation and updating during compile-time optimizations, within an optimizing compiler, comprising the steps of:

(1) accessing a first intermediate representation of the source code of a computer program, wherein said first intermediate representation includes instructions instrumented into the source code of said computer program;

(2) annotating said first intermediate representation with previously-gathered feedback data from a plurality of sample executions of said computer program;

(3) updating said feedback data according to a pre-defined propagation scheme;

(4) performing an optimization of said first intermediate representation annotated with said feedback data updated in step (3), thereby producing a transformed intermediate representation; and

(5) repeating steps (3) and (4) at least once;

whereby the compiler produces more efficient executable program code from said first intermediate representation, thus speeding up the execution of said computer program.

2. The method of claim 1, wherein step (4), comprises the step of performing at least one of the following optimizations:

(i) dead code elimination;

(ii) dead store elimination;

(iii) branch elimination; and

(iv) code transformation.

3. The method of claim 1, wherein said first intermediate representation is a tree corresponding to a procedure within the source code of said computer program.

4. The method of claim 3, wherein step (2), comprises the steps of:

(a) constructing a control flow graph from said tree; and

(b) annotating a frequency value to an edge of said control flow graph, wherein said frequency value corresponds to the number of times that said edge was traversed during said plurality of sample executions of said computer program.

5. The method of claim 4, wherein said frequency value annotated to said edge of said control flow graph is one of the following:

(i) EXACT;

(ii) GUESS;

(iii) UNKNOWN;

(iv) UNINIT; and

(v) ERROR.

6. A computer program product comprising a computer usable medium having computer readable program code means embodied in said medium for causing an application program to execute on a computer that performs precise feedback data generation and updating during compile-time optimizations, within an optimizing compiler, said computer readable program code means comprising:

first computer readable program code means for causing the computer to access a first intermediate representation of the source code of a computer program, wherein said first intermediate representation includes instructions instrumented into the source code of said computer program;

second computer readable program code means for causing the computer to annotate said first intermediate representation with previously-gathered feedback data from a plurality of sample executions of said computer program;

third computer readable program code means for causing the computer to update said feedback data according to a pre-defined propagation scheme;

fourth computer readable program code means for causing the computer to perform an optimization of said first intermediate representation annotated with said feedback data updated by said third computer readable program code means, thereby producing a transformed intermediate representation; and

fifth computer readable program code means for causing the computer to re-execute said third and fourth computer readable program code means at least once;

whereby the compiler produces more efficient executable program code from said first intermediate representation, thus speeding up the execution of said computer program.

7. The computer program product of claim 6, wherein said first intermediate representation is a tree corresponding to a procedure within the source code of said computer program.

8. The computer program product of claim 7, wherein said second computer readable program code means comprises:

sixth computer readable program code means for causing the computer to construct a control flow graph from said tree; and

seventh computer readable program code means for causing the computer to annotate a frequency value to an edge of said control flow graph, wherein said frequency value corresponds to the number of times that said edge was traversed during said plurality of sample executions of said computer program.

APP A47